Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

3-2017

Collaboration trumps homophily in urban mobile crowdsourcing

Thivya KANDAPPU
Singapore Management University, thivyak@smu.edu.sg

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

TANDRIANSYAH

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Computer Engineering Commons, and the Software Engineering Commons

Citation

1

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

2-2017

Collaboration trumps homophily in urban mobile crowd-sourcing

Thivya KANDAPPU
Singapore Management University, thivyak@smu.edu.sg

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Randy Tandriansyah DARATAN
Singapore Management University, rtdaratan@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Databases and Information Systems Commons, and the Software Engineering Commons

Citation

KANDAPPU, Thivya; MISRA, Archan; and DARATAN, Randy Tandriansyah. Collaboration trumps homophily in urban mobile crowd-sourcing. (2017). *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, Portland, Oregon, USA, 2017 February 25 - March 1*. 902-915. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/3629

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email library@smu.edu.sg.

Collaboration Trumps Homophily in Urban Mobile Crowd-sourcing

Thivya Kandappu, Archan Misra, Randy Tandriansyah
School of Information Systems, Singapore Management University
{thivyak, archanm, rtdaratan}@smu.edu.sg

ABSTRACT
This paper establishes the power of dynamic collaborative task completion among workers for urban mobile crowd-sourcing. Collaboration is defined via the notion of peer referrals, whereby a worker who has accepted a location-specific task, but is unlikely to visit that location, offloads the task to a willing friend. Such a collaborative framework might be particularly useful for task bundles, especially for bundles that have higher geographic dispersion. The challenge, however, comes from the high similarity observed in the spatio-temporal pattern of task completion among friends. Using extensive real-world crowd-sourcing studies conducted over 7 weeks and 1000+ workers on a campus-based crowd-sourcing platform, we quantify the effect of such "task completion homophily", and show that incorporating such peer-preferences can improve worker-specific models of task preferences by over 30%. We then show that such collaborative offloading works in spite of such spatio-temporal similarity, primarily because workers refer tasks to their close friends, who in turn perform such peer-requested tasks (with over 95% completion rate) even if they experience detours that are significantly larger (often more than twice) than what they normally tolerate for platform-recommended tasks.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous.

Author Keywords
crowd-sourcing; collaboration; social-ties; homophily

INTRODUCTION
Mobile crowd-sourcing, whereby a time-varying pool of voluntary workers perform *location-specific* micro-tasks, has rapidly become a powerful paradigm for many urban services, such as last-mile package delivery (e.g., Amazon Flex[1]) and municipal monitoring (e.g., Apps such as OneService[2] or NYC311[3] that allow reporting of problems related to garbage, potholes, etc.).

[1]http://flex.amazon.com
[2]http://www.mnd.gov.sg/mso/mobile-installation.htm
[3]http://www1.nyc.gov/connect/applications.page

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
CSCW '17, February 25–March 01, 2017, Portland, OR, USA.
© 2017 ACM. ISBN 978-1-4503-4335-0/17/03 . . . $15.00
DOI: http://dx.doi.org/10.1145/2998181.2998311

While such participatory models of service execution have several advantages, most crowd-sourced services currently suffer from less-than-optimal resource utilization. In particular, research has shown((e.g. [5]) that *centrally-coordinated* and *predictive* models of task recommendation (that take into account the future movement trajectory of individual workers) can significantly outperform current *decentralized* and *myopic* models, where each worker greedily and independently chooses the closest set of available tasks. While such predictive, central coordination promotes higher task completion and higher worker productivity (lower detour overheads), the overall task completion rates can still be low–in many cases, workers often deviate from their expected travel paths and fail to complete tasks that they previously accepted. The challenge of task completion is exacerbated for *task bundles*, where workers must complete multiple tasks to receive payment; it is worth noting that task bundling is a commonplace practice in many urban crowd-sourcing platforms (e.g., package delivery), as the higher cumulative reward helps overcome the usually low per-task payment rates.

Developing and validating mechanisms to improve overall task completion rates is thus an important challenge for urban mobile crowd-sourcing. In this paper, we explore a central idea, namely **dynamic collaboration among peer crowd-workers**, and carefully investigate whether and how such a feature can help to significantly improve the overall task completion rate. To empirically investigate this idea, we utilize *TA$Ker*, an experimental campus-scale crowd-sourcing platform (with a client-side mobile App for both Android & iOS devices) that has been operationally deployed on our Singapore Management University (SMU). *TA$Ker* has an active crowd-worker base of over 1200 university students, who earn real monetary rewards by performing a variety of location-specific *reporting* tasks, such as "length of queue in the food court", "availability of a soda brand at a particular vending machine" and "cleanliness level of a specific toilet".

We assume that dynamic collaboration would work as follows: worker A (the *nominator*), who has previously accepted a task X but finds herself unable to complete it in the stipulated time (most likely because the task location no longer lies on A's commuting path) can choose to *refer* the task to a *nominee*: a peer co-worker B, who *may* be closer to task X. If B completes the task within the stipulated time, then A and B share the resulting reward. Note that current mobile crowd-sourcing platforms do not possess such real-time peer-to-peer task referral features.

Through our experimental studies, we demonstrate our central contribution, namely that such a referral feature is indeed *very effective* in increasing task completion rates. But most interestingly, this increase is NOT because of the likely proximity of the nominee to the task in question. In particular, we shall discover a phenomenon called *task completion homophily*: workers with strong strength-of-ties tend to have high similarity in their spatio-temporal patterns of task execution. In other words, birds of a feather flock together. Consequently, a referral to a friend co-worker may have limited utility, as tasks that are out-of-the-way for person A are also more likely to be distant from the travel path of A's friends. Instead, the increased effectiveness of such referrals arises because *peer-requests* for task completion exhibit much higher compliance than platform-generated recommendations. Tellingly, nominators refer task to their closest friends, and nominees in turn complete such peer-referred tasks even if they incur significantly higher detour overheads.

Research Questions & Contributions: We empirically and systematically address the following research questions:

- *Are there any underlying influences of social-ties on task completion patterns–i.e., do the task completion patterns of worker "friends" exhibit higher spatio-temporal similarity?* Using strength-of-tie metrics inferred from the physical-world movement and interaction of *TA$Ker* workers on the university campus, we show that there is a strong correlation (Pearson correlation value of 0.74) between the tie-strength of a worker pair and their spatio-temporal task completion patterns. Our results indeed reveal the so-called 'task completion homophily' phenomenon described earlier.

- *Is it possible to better predict the spatio-temporal preferences of tasks for a mobile crowd worker, by considering the corresponding preferences of other peers–i.e., how does the prediction accuracy of task completion patterns depend on social factors?* We first show that a user's intrinsic preferences for selecting and completing tasks can be learnt and modeled, using various crowd-sourcing-related features such as (i) the task's detour overhead, (ii) the popularity of the task location and (iii) the task's incentive (with the last two factors having a dominant effect on worker preferences). Finally, we show that incorporating social factors (specifically, the strength-of-tie weighted location preference of all other peer workers) in a per-user task preference model results in a significant improvement, increasing the prediction accuracy by 31%.

- *Does dynamic collaboration- (i.e., peer referrals) result in any tangible benefit to the overall task completion rate?* By incorporating such a peer referral feature in the *TA$Ker* platform, we demonstrate its popularity and effectiveness. Such a peer referral capability increased the task completion rate by 14% (from a baseline rate of 24.4% in the absence of such referral capability). For bundled tasks (especially if the bundle is spatially dispersed), the effect is even more dramatic: the completion rate of referred tasks more than doubles! Moreover, even over a relatively short 1 week trial period, this peer referral feature was utilized by 10% of active *TA$Ker* workers to offload approx. 300 tasks ($\approx$ 5.3% of all tasks performed during that period).

- *How do workers distribute such referrals among their "friend" workers? And how exactly do peer referrals result in higher task completion rates?* We carried out a detailed analysis of the task offloading pattern of *nominators* and the task acceptance/completion pattern of *nominees.* We found that the workers *always* offload tasks to their strongest ties, trusting that their closest friends will complete the task. We then show that (i) as expected, nominators refer/offload tasks that are likely to result in the largest detour (the detour overhead for referred tasks is roughly 1.8-times the detour experienced for non-referred tasks), and (ii) nominees exhibit high task completion rates, with even detour-sensitive workers performing tasks that impose a detour overhead of > 10 mins (even though they accept system-recommended tasks only if the detour is much smaller (< 5 mins)). Moreover, the average detour overhead experienced by a nominee for such offloaded tasks is strongly correlated (Spearman correlation coefficient=0.64) to the tie-strength between the (nominator, nominee) pair.

Overall, we believe that our work is the first to empirically demonstrate that a 'peer referral" capability can significantly increase the task completion rate of mobile crowd-sourcing, by utilizing the willingness of workers to indulge in "social collaboration", in spite of higher individual travel overheads. While our studies are restricted to a university campus, we expect this 'referral' feature to be effective in city-scale crowd-sourcing platforms as well.

RELATED WORK

Research looking at computer supported cooperative work has long recognized the value of collaboration for completing tasks distributed over a network [11, 18]. Below we review the literature in modeling user behaviour, understanding the value of collaborative work and how social connections affect individual and collective movement patterns.

Benefits of Collaboration

A wide body of literature attests to the benefits of collaborative problem-solving in various domains. For example, pairs were shown to outperform individuals in lab tasks simulating scientific discoveries [23]. Co-authored scientific papers are cited more often and appear in more prestigious journals than single-authored papers [10, 24, 29]. Teams developing inventions create more influential patents and fewer very poor patents [27]. However, not much is known about the underlying social ties between crowd workers and how social-tie based interactions among workers may be leveraged to improve mobile crowd-sourcing outcomes.

Collaboration in Crowd-sourcing

More recently, crowd-sourcing research has turned its attention back to the value of coordinating human interaction at scale. Some of this work is informed by the limits of leaving individuals to self-organize their contributions to larger projects [26]. Since the existing crowd-sourcing platforms (both online and mobile) have no built-in way for workers to communicate each other, the platforms believe that workers do not communicate with each other as part of their work unless the platform is engineered to facilitate it. There is also a growing recognition of tangible benefits of incorporating what Huang has recently proposed "social facilitation" [8, 12, 13] and the value of group identity to collaboration [31]. "Friend-sourcing", for example, [2] offers a valuable case study for incorporating one's social network into crowd-sourcing processes. While [2] identifies the potential issues around friend-sourcing and collaborative work, they conclude that the increase in output quality outweighs the potential challenges. Many of the studies examining the value of integrating social networks into crowd-sourcing to facilitate collaboration between sets of workers rather than making room for collaborations that workers develop themselves [4]. In contrast to this research that primarily looks at online crowd-sourcing scenarios, we focus on improving the task completion rate for location-based mobile crowd-sourcing (where collaboration is subject to additional spatiotemporal constraints as workers must be at the task location within a predefined time window).

Task Acceptance & Completion Behavior

Limited studies have explored the relationship between task attributes and worker behavior, especially for the crowd-sourced execution of location-dependent tasks. Wang [30] studied the task completion times of online tasks (posted on Amazon Mechanical Turk), and established a power-law relationship between task completion times and task-related features, such as the type of the task, the task price and the day the task was posted. Alt et al. [1] used an independently developed mobile crowd-sourcing platform to discover a variety of worker preferences, including preference for performing tasks before and after business hours or involving relatively simple chores (e.g., taking pictures). More recently, Thebault-Spieker [14] conducted studies on the relationship between task pricing and location, at city-scale, and showed that workers preferred to perform tasks with lower detours and that were outside economically-disadvantaged areas.

How Social Ties Influence Physical World Movement

There is a rich body of work in social computing on how an individual's movement pattern is modulated by the movement choices of her friends. Cho et al [7] used location-based social network data and cellular location records to demonstrate that social relationships (in particular the visit pattern of friends) explain around 30% of human movement (especially over longer distances). The correlation in the movement behavior of friends was also utilized by DeDomenico, et al [9] to show that a user's movement behavior could be more accurately predicted by factoring in the movements of her friends. In our work, we carefully study how this concept of social influence interacts with the prospect of collaborative task execution in mobile crowd-sourcing platforms.

BACKGROUND AND OVERVIEW

Before presenting the design of the *TA$Ker* platform, and the associated experimental results, we first provide a high-level background of the deployment of our mobile crowd-sourcing application on our urban university campus.

Campus Layout: SMU has a downtown campus in Singapore, consisting of 4 distinct academic buildings, 1 library, plus 1 administrative building that is rarely utilized by students. The student body comprises approximately 9000 undergraduate students pursuing degrees across 6 schools, who typically commute daily from their home to the university. Each of the 4 academic buildings is 5 storeys in height, with a per-floor area varying between 1500-2500 m^2, and consists principally of faculty offices, research labs, teaching classrooms and various "group-study rooms". The library is 4 floors in height, with a per-floor area of approx. 2750 m^2.

Moreover, all the 5 buildings are connected by an underground, airconditioned concourse (accessible from each building's basement level). This concourse is the "social hub" of the campus and includes a variety of eating establishments, a bank, multiple retail shops and benches for studying, as well as serves as a common space for a variety of extra-curricular recreational activities (due to Singapore's hot and humid climate, inhabitants usually spend majority of their working day indoors and typically use the underground concourse to transit between buildings). See Figure 1(a) for a schematic layout of the campus, and Figure 1(b) for images of certain popular sections of the underground concourse. The food court, and nearby library areas, are the most heavily trafficked, and the bench areas are usually occupied by students for moderately long durations.

(a) Schematic of the SMU campus (buildings & concourse)

(b) Various popular sections (campus clinic, food court, T-Junction for events, study areas) of the concourse

Figure 1. The urban campus: Venue for *TA$Ker*

Location Service and Student Engagement: As part of a large-scale experimental mobile testbed [19] (which presently has a participating pool of over 3000 undergraduate students), the university operates an indoor location service [16]. This location service does not require any application installation on any mobile device, and instead uses server-side Wi-Fi fingerprinting techniques to track the on-campus location of all persons (i.e., their mobile device) on campus, as long as their Wi-Fi interface is enabled (regardless of whether they connect to the campus Wi-Fi network or not). Due to well-known limitations of such server-side location tracking, the location service currently offers medium-grained granularity (errors are typically $\pm 6-8$ meters) and latency (the period between successive location updates about individual devices is around 2-4 minutes). However, the universality of this location service means that we can perform continuous, longitudinal tracking of all mobile devices on campus. As a consequent, the location service provides the location/movement history of over 100,000 devices over the past 2.5 years; this movement data can be used predict an *TA$Ker* user's individual future movement pattern.

What Do We Crowdsource?: *TA$Ker* is built and deployed as an *experimental mobile crowd-sourcing platform*, which can support empirical investigation of novel and alternative crowd-sourcing strategies over a campus-scale deployment. At present, the worker pool of *TA$Ker* comprises around 1000 students (a subset of the pool of 3000 student participants of the campus testbed). In addition, the *TA$Ker* App is intended to help the Facilities Management (FM) services on campus achieve the vision of a *smart campus*, by providing continual, updated information on the state of various resources on campus. Examples of such resources are varied and include: the cleanliness of different restrooms, whether the lights in certain rooms have been switched off or not, the crowdedness of certain study areas, the availability of specific items in various vending machines and whether various garbage bins are full or not. Accordingly, all tasks in *TA$Ker* are presently *reporting-type* tasks, with the goal of using the university's student population as a voluntary and participatory resource to provide reliable and timely inputs on the state of such campus resources.

SYSTEM DESIGN

Before diving into the details of the experimental studies that are relevant to this paper's central research question, we first outline the key relevant characteristics of *TA$Ker*. Figure 2 illustrates the overall functional architecture of *TA$Ker*. In addition to the *TA$Ker* mobile App, the *TA$Ker* backend system includes the following key components: (a) *Route Predictor*, that predicts an individual user's movement trajectory based on her historical movement traces (the traces obtained by the server-side Wi-Fi location tracking system); (b) *Task Recommender*, that, for a certain pool of workers, suggests tasks (from the overall pool of available tasks) that best match (i.e., minimize the additional detour overhead) the predicted trajectory of an individual worker; (c) *Task Management Portal*, that allows *TA$Ker* administrators to create, modify and monitor the set of available tasks (as well as set task prices or specify task *bundles*), (d) *Database & Results Validator*, which stores the responses received from the users and validates the integrity of the responses by comparing the location of the specified task and the locations visited by the worker, and (e) *Results Analyzer*, which analyzes the contents of executed tasks to provide deeper understanding of the behavioral interaction by crowd-workers with the *TA$Ker* App.

Task Recommendation & Push vs. Pull Users: One of *TA$Ker's* core capabilities is the ability to support customized recommendations for individual workers, by going beyond such proximity filters (which operate using only the worker's *current* location) to preferentially suggesting tasks that lie close to a person's predicted *future movement trajectory*. More specifically, *TA$Ker* allows workers to be randomly divided into two groups: (a) the "pull" class workers are able to view all available tasks and can select an appropriate subset of tasks from this entire list; whereas (b) the "push" class users are able to view only a smaller set of tasks, selected by the *Task Recommender* that are best aligned to their predicted movement pattern. For this "push" class, the Task Recommender utilizes the previously-proposed TRACCS framework [6], which recommends tasks so as to maximize the total set of feasible tasks, while adhering to (i) a worker-specific detour bound and (ii) stochastic uncertainty in a worker's movement trajectory. Broadly speaking, TRACCS improves the decentralized or first-come-first-served models of *task assignment*, a topic that Kittur et al. [17] have identified as critical to the sustainable operation of future crowd-tasking platforms.

Implementation Details

The implemented *TA$Ker* system consists of three components: (a) a web interface for task creation, (b) a server and a database for storing tasks and responses, and (c) a client application on mobile phones for crowd-workers. The PHP-based server implements the *Route Predictor*, *Task Recommender* and *Task Management Portal* (creation and modification of tasks), and also allows individual workers to be assigned to different control/treatment groups (e.g., push vs. pull workers). We developed both Android and iOS mobile client Apps for *TA$Ker*, and distributed them to our participants via our own private App Store. This *TA$Ker* App shows various tasks to the users which can further be filtered based on users' current location, incentive preferences, preferred task types, etc., and allows them to select and execute tasks, with the results then being uploaded to the server.

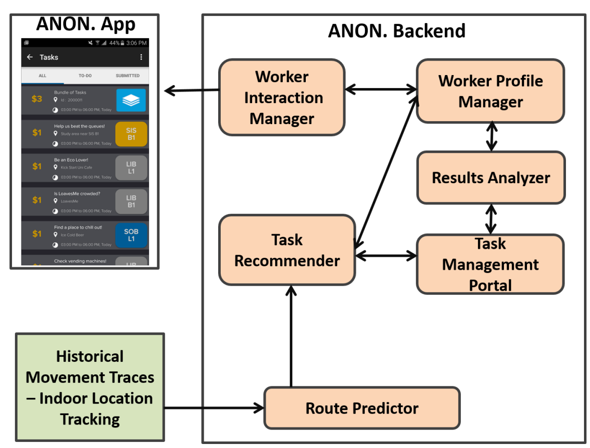
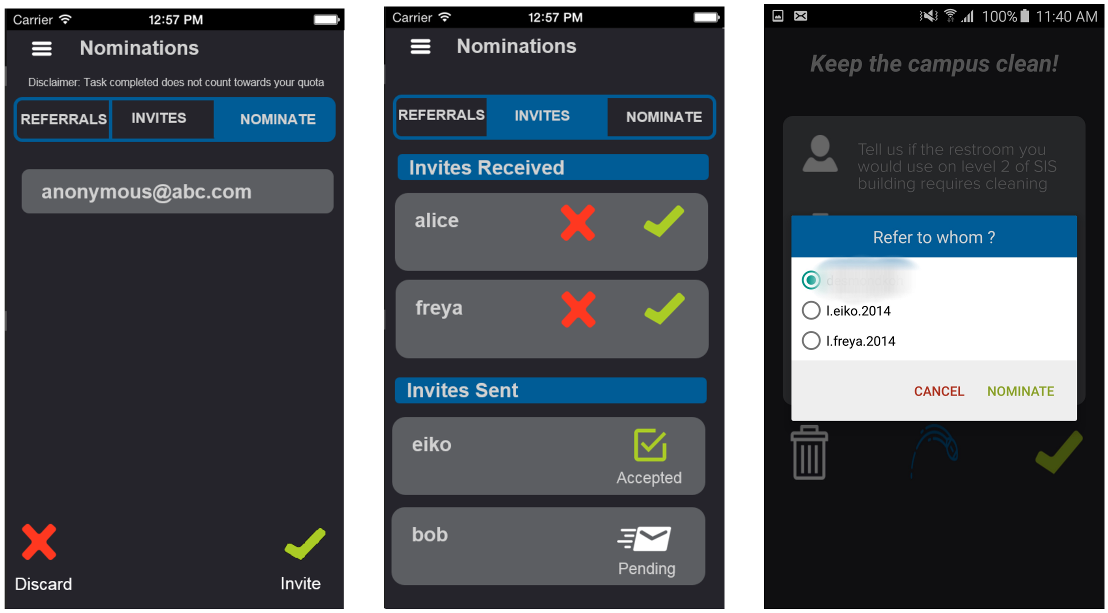


Figure 2. *TA$Ker* framework - architecture

Experiment Study Details

To support the studies, on the interplay between social ties and peer recommendations, performed in this paper, *TA$Ker* was deployed over a 7 weeks period, March 7 - April 22, 2016 (with tasks being available only on working weekdays). During this period, a total of 1300 students opted to participate in the study (been previously approved by the university's Institutional Review Board (IRB)). During this period, the *TA$Ker*



(a) Inviting a friend. (b) List of invites sent and received. (c) List of nominees to offload a task.

Figure 3. Screenshots of *TA$Ker* App showing: (a) inviting a friend, (b) list of invitations sent and received, and (c) the process of task offloading to a nominee.

platform provided a total of 140,000 distinct tasks. The users were divided randomly into two equal sized-groups "push" vs. "pull" groups.

For operational reasons, the tasks were broken up into 3 distinct *time windows*: 9am-12noon, 12noon-3pm and 3pm-6pm, roughly aligned with the most common lecture slots, such that the *TA$Ker* App only shows tasks that are valid within the current time window. The *Task Recommender* operates on each window independently–i.e., it takes all tasks within a 3-hour time window, computes the worker's predicted movement over that 3-hour window and then recommends suitable tasks. Moreover, each task is associated with a variable *execution interval* (T_s, T_e) (where T_s and T_e denote the start and end time instants of the interval), such that the task can only be performed (i.e., the report on the relevant resource be uploaded) within this execution interval. To ensure fairness in terms of workloads and rewards among the "push" and "pull" workers, a single worker could only execute *Max* tasks in any 3 hour window (*Max* was set to 6 for all the weeks).

Table. 1 highlights the overall distribution of tasks during this study, including the numbers of individual/same-building (type 1)/multi-building (type 2) bundles. Table. 2 shows the total number of *TA$Ker* workers involved in each week – both registered and active workers and the number of responses received, on weekly basis.

During this 7-week period, we intentionally utilized two novel features of *TA$Ker*, described next.

Task Bundling: In this study, we introduced the notion of task *bundles*: a set of tasks grouped together and offered to the crowd workers at slightly lower per-task incentive (compared to the atomic tasks). The key feature of the bundle is that it provides a higher total reward, and potentially allows a worker to amortize their total detour overhead over multiple tasks (especially if the tasks are all close to one another). The downside, however, is the absence of any partial rewards: a worker receives the total payment specified for the bundle only if she completes *all* the tasks (there are no partial rewards for performing a subset of the tasks in a bundle).

Table 1. Summary of task details.

	Indiv. tasks	Tasks in type1 bundle	Tasks in type2 bundle
Incentive per task ($)	0.40	0.30	0.35
Total tasks posted	37.45%	27.88%	34.67%
Posted tasks per building			
SIS	6.31%	4.6%	
SOB	9.07%	7.3%	
LIB	6.61%	2.77%	
SESS	9.7%	8.01%	
SOA	5.76%	5.22%	
Non-referred task completion rate	66.94%	11.93%	21.11%
Referred task completion rate	79.55%	100%	52.17%

Our bundling technique is motivated by several key discoveries and practical use cases: (i) Recent studies [21, 22] of a year-long dataset from a leading mobile crowd-sourcing platform point out an important factor behind the success of a small set of efficient workers (called *super agents*), who constitute 10% of the crowd but perform 80% of the tasks. These efficient workers optimize their efficiency by carefully plan-

Table 2. Summary of user details.

Week	No.Registered users	No.Active users	Responses received
1	1119	349	10618
2	1167	364	10624
3	1201	308	8491
4	1208	262	8301
5	1222	248	8936
6	1231	223	6779
7	1235	171	5641

ning their trips, and bundling a sequence of tasks to perform in a single trip; (ii) Several real-world crowd-sourced package delivery companies in Singapore work on the bundling model: to compensate for the low per-delivery task rewards associated with their low-margin operations, they need to provide each worker with multiple pick-up/delivery jobs.

Moreover, by varying the *spatial dispersion* of the bundle, we can effectively control the amortization benefit. For example, if the tasks are all on the same floor of one of the academic buildings, they can be performed with a detour of less than 5 minutes; in contrast, a bundle spread across academic buildings could potentially incur a travel detour of as much as 15 minutes. More specifically, our studies included 3 different types of tasks: (1) individual tasks – where worker will get paid after completing it, (2) Same-building bundles – where 4 tasks in the same building are grouped together and formed as a bundle, and these bundles are posted to cater people who do not want to make longer detours, and (3) Multi-building bundles – tasks belong to this type of bundle are distributed across the campus, covering any 4 of the 5 buildings.

Friend Nominations & Task Referrals: To improve the completion rate of tasks, and to experimentally study the impact of "peer requests" on the task execution pattern, we introduced the "task referral" feature. Under this feature, each worker could, a priori, invite at most 4 of his/her co-workers (existing *TA$Ker* workers) to become "*TA$Ker* buddies". If this request was accepted, the referring worker could then refer/offload an accepted, but currently not executed, task to such a buddy. (Within a single 3-hour time window, a *TA$Ker* worker was allowed to offload at most 4 (out of the maximum of 6) tasks to such buddies.) If the offloading request was accepted, then the task could no longer be performed by the referrer, but only by the requested buddy. Moreover, if the requested buddy completed the task, referring worker will receive 25% of the task incentive as a *bonus* while requested buddy will receive the task incentive.

The idea of task offloading emerged from our experiences with an earlier version of *TA$Ker*, which was used to conduct a pilot study over 900 workers in Sept 2015. From that study, we observed that 15% of the accepted tasks are not completed by the crowd workers. Moreover, from an online survey conducted on those workers, 81% workers answered that *the primary reason behind their failure to complete a previously-accepted task was a sudden change in their on-campus movement pattern*. Such unexpected changes caused some of their tasks to lie too far from their actual trajectory pattern. By allowing a worker to dynamically co-opt another worker, we expect the task referral feature to help mitigate this problem.

Due to the delay in completing and integrating this task referral feature, this referral feature was activated only during the last week (happen to be the end of semester exam week in the campus) of our study. Note that the process of becoming a "*TA$Ker* buddy" is unidirectional. In other words, if a *nominee* B accepts a buddy request from a *nominator* A, the nominator can refer tasks to the nominee, but not vice versa. For the reverse flow of task referrals (from B to A), a separate and explicit buddy request must first be initiated (by B) and accepted (by A). Fig. 3(b) illustrates screenshots of the *TA$Ker* App, during various stages of the buddy request and task referral processes.

TASK COMPLETION & SOCIAL TIES

We first start our investigation by trying to understand the "homophily effect" in mobile crowd-sourcing – i.e., we specifically address the research question *"Do closely-connected peer workers exhibit spatio-temporal similarities in terms of the tasks performed (e.g., whether they perform tasks at the same or nearby locations, or during similar time periods)?"* The overall goal is to understand whether such social similarity effects influence worker behavior, even in situations where the crowd-sourcing platform does not natively (or intrinsically) support collaborative task execution.

To investigate this issue, we first compute the expected strength-of-tie among pairs of *TA$Ker* workers using just their physical collocation and interaction patterns. Having established such a physical world movement-based proxy for tie-strengths, we then study whether and how the spatio-temporal similarity of task execution is related to such a tie-strength metric.

Calculating Pairwise Strength of Ties

To compute the pair-wise tie-strength between any two *TA$Ker* workers, we utilize the historical traces of worker location on the campus, and combine that with prior work on (i) movement-based group detection and (ii) movement-based tie-strength computation. We define a spatio-temporal co-occurrence of two (or more) users as an *episode*. We then utilise the longitudinal observations we are able to make using the location traces data, of extracted episodes, to build a pairwise tie-strength metric.

More specifically, we first utilize the state-of-the-art *GruMon* group detection system [25] to detect group interactions, based on shorter time windows of shared residency (at different locations) and movement patterns. Given our access to only location data traces (we do not have the sensor data streams of each worker's smartphone), we utilize *GruMon's* location-based group detection logic, which looks for joint transitions between distinct *stay points* (locations where the user resides for a substantial amount of time) to reliably infer group interactions. Prior studies have shown that *GruMon* has accuracies of approx. 90% and is fairly robust to location errors. We then also compute a combination of features (involving both collocation and group interaction) to compute a pairwise tie-strength among individuals, using the approach previously adopted in [15], which combines the following factors:

- *Spatial precision:* the inverse of the number of other people the meeting/episode is shared with. For example, the strength between two people in the group study room is higher than that of two people who are in a seminar room with 50 others.
- *Temporal precision:* represents the normalized duration of meeting/episode. Longer the duration, stronger the tie.
- *Spatial uniqueness:* For example, meetings at the Gym are more rare/unique than at the seminar room, hence higher weights are given to the former. This is computed based on ALL meetings observed over the period of consideration.
- *Temporal uniqueness:* Meetings that occur during the weekends/late evenings are given a higher weightage over meetings during the day, as they are more uncommon.
- *Durability:* this gives higher weights to pairs of people who meet multiple days over the week (their relationship has longevity)
- *Frequency:* This feature computes the number of distinct group interactions between a pair of individuals during the day, and thus gives higher weightage to people who meet multiple times over the same day.

We retain only the top 1% of strongest ties so as to prune out accidental co-locations (i.e., strangers being at the same place at the same time accidentally).

Validation: The demographic similarity (school and year of study) between the top-100 inferred (strong) pairs and 100 random pairs were compared. Using a Chi-squared test, we confirmed that the strong pairs showed a statistically, significantly higher similarity. In other words, stronger ties were seen to disproportionately become to the same school, or the same cohort, which is indeed what we expect. For the purpose of this study, the strength-of-ties among individuals was computed using the entire history of indoor movement data from the period of September 2015 through February 2016.

Threat to Validity: While the movement history from Sept 2015-February 2016 provides sufficient data to build stable tie-strengths, the model admittedly assumes that these tie-strengths remain constant over the observation window. A particular concern might be that this observation period straddles two distinct academic terms; one can expect many relationships to change from term to term (e.g., ties between students sharing the same course or working on the same project). However, the top 1% of ties typically represent the long-lasting, enduring friendships–these ties were seen to remain unchanged when the tie-strength computation was done separately for each term. Accordingly, we believe that these top-1% ties provide a fairly robust indicator of relationships among the student workers using *TA$Ker*.

Spatio-temporal Task Completion Score

We next observed the task completion pattern of each pair of *TA$Ker* users, over a seven-week trial period (collecting details such as (i) time that task is performed; (ii) number of coordinated transitions across locations over discrete time intervals (e.g., 15 minutes)). Given these observations, we defined two distinct *Task Similarity Scores* as follows:

1. *Collocation-Time Similarity Score:* For every 15 minute time window over the period of observation, we extract the locations in which a pair of workers complete tasks. For example, assume user A and B are known to have a strong tie strength and user A completes tasks at 3 locations {campus clinic, food court, library} between 11:15 and 11:30. Also, User B completes tasks at {food court, concourse, library} during the same time window. In this case we define user A and B as sharing 50% of their spatio-temporal profile, i.e., 2 (i.e., food court and library) out of the 4 (i.e., food court, library, campus clinic and concourse) distinct locations observed over that time window. The aggregated score is first calculated, pairwise on daily basis, and averaged over the entire period of observation.

2. *Collocation Similarity Score:* This method is similar to the Collocation-Time Similarity score, except that we look at the shared task-completion locations over an entire day– i.e., without requiring that a shared location tasks completed by both users during the same 15-minute interval. In other words, given two distinct sets of locations $A(T)$ and $B(T)$ where users A and B execute tasks over a day, this score computes $\frac{A(T)\cap B(T)}{A(T)\cup B(T)}$.

"Homophily" Effects in *TA$Ker*

We next investigate the correlation between the tie-strengths (based on longitudinal observations of collocation and movement patterns) and the spatio-temporal task completion scores. Fig. 4 illustrates a scatter-plot of the two distinct scores (Collocation-Time and Collocation) vs. the Tie-Strength scores between all pairs of *TA$Ker* participants. It is fairly easy to see that the Spatio-temporal scores are indeed positively correlated with the tie-strength values. Furthermore, we calculated the correlation between both the spatio-temporal score and the corresponding tie-strength values: the correlation coefficient is 0.53 (with p-value 1.99e-15) for Collocation Similarity and 0.74 (with p-value 2.2e-16) for Collocation-Time Similarity. Clearly, "friends" perform tasks not only at the same locations, but during the same 15-min time windows.

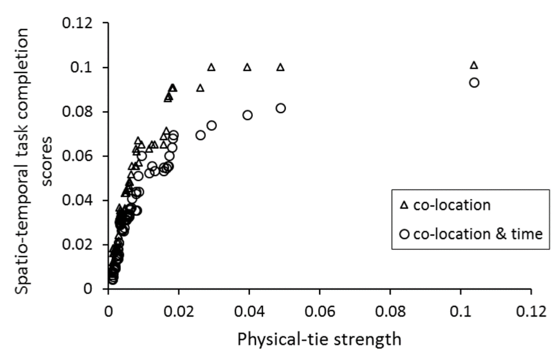


Figure 4. Tie strength between pairs of *TA$Ker* participants

Key Takeaway: Our results show the existence of a significant "homophily" effect in the execution of mobile crowd-sourcing tasks on campus. Workers who share the same school or belong to the same cohort not only have closer "social" ties to one another, but also exhibit similarities in the *spatio-temporal preferences* for executing tasks. This is likely due to the fact that such stronger-tie workers ("friends") have a higher likelihood of having similar movement patterns (e.g., possible common courses, highly overlapping course schedules), and thus end up preferring tasks with similar space-time distributions. At first glance, this phenomenon suggests that allowing explicit collaboration/referral of tasks among "friends" may not be as useful, as their travel patterns may be similar (birds of a feather truly flocking together), resulting in similar likes/dislikes for tasks at different locations. We shall revisit this issue later, when we carefully analyze the outcomes that result from explicit support of task referral in the *TA$Ker* platform.

WORKER PREFERENCES & SOCIAL INFLUENCE

Workers may have different preferences and strategies for accepting and executing tasks on a mobile crowd-sourcing platform. In general, such individual-specific preferences can be affected by *platform-intrinsic* factors such as incentives, the detour overhead and the task's complexity (the time taken to perform the task). Continuing with our investigation of the possibility of homophily effects on mobile crowd-sourcing, we first develop a per-user model of task preferences based on platform-intrinsic features. Subsequently, we incorporate the *social effect*–i.e., taking into account the preferences of other peer-workers with whom the worker has strong social ties, and show that such social features can significantly improve the prediction of individual worker preferences. (Besides being germane to our investigation of how such social ties affect collaborative crowd-sourcing, such preference modeling can be used to improve the operational efficiency of the platform in other ways–e.g., by improving the relevance of tasks recommended by the *Task Recommender* component).

In the following we first explain how to parameterize a worker's behaviour. Then we explain our approach for modeling the worker context over time. Let us denote a worker's likelihood of completing a new task by $\mathcal{P}(y|x;\theta)$, where variable y indicates whether the worker completes the task successfully $(y=1)$ or not $(y=0)$, vector $x=(x_1,x_2,...)$ includes the parameters (such as payment, detour, task complexity, etc.) and vector $\theta=(\theta_1,\theta_2,...)$ is the learned coefficients based on worker's history.

Parameterizing Worker's Behaviours

We first model θ (the worker's preference) as a function of the following platform (or task) *intrinsic* features:

- *Incentive:* the amount of money the worker would receive by doing a task. This is specified by the task owner via the *Task Management Portal*.
- *Detour:* this quantifies the additional travel overhead that the worker incurs while performing a task. To measure the detour, we first need to identify the neighboring *stay locations* (both prior to and after the task performance)–i.e., locations where the worker stays for a significant amount of time (in our case, more than 4 minutes). The detour is then calculated by computing the difference between the shortest travel path between these stay locations and the path the worker actually takes so as to perform the chosen task. Mathematically, let the task location be denoted as Z. We analyze the location traces, and identify the stay locations (places where the worker resided continuously for 4 or more minutes) before and after going to Z. We denote the stay locations before and after Z as X and Y, respectively. The detour time is then $(t_{X,Z}+t_{Z,Y})-t_{X,Y}$, where $t_{X,Z}$ denotes the travel time to reach location Z from location X.
- *Task complexity:* this is a relative metric to measure the complexity of tasks. In *TA$Ker*, there were 4 different *types* of reporting tasks, ranging from ones that required choosing from a *Boolean option* to *Photo* tasks (those that required taking a picture). To roughly measure the complexity of each type, we first calculate the daily ratio between the average time the worker spent on tasks of that type, divided by the average time he/she spent on all completed tasks that day. The final complexity is then obtained by averaging this daily ratio over the entire 7-week observation period. As an example, let the amount of time spent while performing *photo* type tasks on day i be denoted by by $t_{ph}(i)$, total time spent on performing tasks on that day by $t_{all}(i)$, and let n be the total number of days. The complexity score for *photo* tasks (for that specific user) is then obtained as: $\frac{\sum_{i=1}^{n}(t_{ph}(i)/t_{all}(i))}{n}$.
- *Task Familiarity:* this is a relative metric used to find out the preferred type of tasks of each user, on a weekly basis. This metric is computed by dividing the number of tasks of a particular type completed by a user by the total number of tasks completed by the same user in a period of time.
- *Popularity of the task location:* number of unique workers (during the specified task validity period, eg., between 10 and 10:30) in that particular location. This measure of a location's intrinsic popularity is then normalized by the worker count over all possible locations across the entire campus.

The training output vector indicates whether or not the worker has completed the task he/she accepted. Given the above parameters, we can run our algorithms to learn θ and consequently capture worker-specific preferences.

Regression Model For Worker Task Preferences

We exploit regularized logistic regression to build the predictive model and find parameter weights (i.e., vector θ). Let us denote the collected observations from a worker by $(x^{(i)},y^{(i)})$ where $1\leq i\leq T$. The regression model aims to find θ by minimizing the following cost:

$$arg\min_{\theta}\frac{1}{T}\sum_{1\leq i\leq T}\log(h_{\theta}(x^{(i)}))y^{(i)} + \log(1-h_{\theta}(x^{(i)}))(1-y^{(i)})^2+\lambda_r\sum_{j=1,2,...}\theta_j^2 \quad (1)$$

where $h_\theta(x) = \mathcal{P}(y=1|x,\theta) = \frac{1}{1+\exp(-\theta' x)}$ is the likelihood of completing a task by the worker and λ_r is a penalty coefficient, which is a small positive number that shrinks the norm of θ. The cost function minimizes the prediction error over the training dataset. The penalty term, also known as Ridge regularization, is used to avoid overfitting the model. To solve Equation 1, we use a gradient descent based algorithm that iteratively converges to the solution [3].

We employed regularized logistic regression with Ridge penalty on the parameters. Each worker completed between 10 – 6000 tasks over the trial period of 5 weeks (as the university end of semester exams were going on for the last two weeks, we have excluded those 2 weeks from this analysis, assuming their decision to choose a task highly depends on their tight schedule). To test the accuracy of the resulting model, we divided the 5 week trial period into two sets: (1) the first three weeks of data (50%) formed the training set, while (2) the next two weeks (4th and 5th) of data (29%) formed the test set. After finding a suitable λ_r corresponding to each worker, we applied the gradient descent based algorithm to find the characterization parameter θ for each worker.

Table 3. Regression Table.

Features	Coeff.	Std.Err	Chi-sq	p-val	95% CI
Intercept	-1.23	0.36	11.14	0.00084	(0.15, 0.61)
Detour	-0.039	0.014	0.08	0.0077	(0.977, 1.03)
Incentive	0.59	0.10	7.60	0.00058	(0.27, 0.57)
Complexity	-0.18	0.23	0.67	0.41	(0.53, 1.30)
Familiarity	0.08	0.022	15.04	0.00010	(0.88, 0.96)
Popularity	0.68	0.18	7.55	2.8e-05	(1.51,3.12)

We tabulate the regression outcome in Table 3. The features *incentive*, *detour* and *popularity of the task location* are deemed to be statistically significant. The regression coefficients also reveal the relative importance of these different platform-intrinsic features. For example, we see that, holding all the other features at a fixed value, we will see a 3.8% increase in the odds of accepting the task for a one-unit decrease in detour (since 1- $\exp(-0.039)$ = 0.038). Similarly, we see an 80% and 97% increase in odds of accepting the task for a one-unit increase in incentive ($\exp(0.59)$ -1 = 0.8) and popularity of the task location (($\exp(0.68)$ -1 = 0.97)), respectively.

Effect of Social Ties on Prediction Accuracy

To measure the effect of social ties, we now include one more factor called *friendship factor* (which measures the collective social preference for a location) as follows. We first derive an aggregated *location score* per each user in our system. This score is calculated by averaging the number of tasks performed in each possible location during the first 3 weeks (i.e., training set) of the trial period. Subsequently, the "friendship" factor for a worker i at a task location l is obtained by computing the weighted average of the "location score" of all other workers (with the weight for worker j being directly proportional to the tie-strength between workers i and j). Mathematically, if TS_{ij} denotes the tie-strength between user i and his peer j and LS_{jl} denotes the location score of the friend j at location l, then the 'friendship" score for user i at location l is computed as $\sum_{j \in Worker \setminus i}(TS_{ij} * LS_{jl}) / |\{Worker \setminus i\}|$.

We tabulate the regression outcome of the new model in Table 4. The 3 older features (*incentive*, *detour* and *popularity of the task location*), as well as the new *friendship* feature are seem to be statistically significant. Similar to the observations we made in the previous model, we can say that, holding all the other features at a fixed value, we will see 136% increase in the odds of accepting the task for a one-unit increase in friend factor (since $\exp(0.86)$ -1 = 1.36).

Table 4. Regression table (with friendship factor)

Features	Coeff.	Std.Err	Chi-sq	p-val	95% CI
Intercept	-0.47	0.17	7.93	0.0048	(0.45, 0.86)
Detour	-0.08	0.022	5.04	0.00010	(0.88, 0.96)
Incentive	0.22	0.30	0.52	0.0004	(0.68, 2.28)
Complexity	-0.011	0.21	1.35	0.82	(0.54, 1.41)
Familiarity	-0.074	0.23	8.01	0.030	(0.84, 0.91)
Popularity	0.47	0.16	4.88	0.0001	(1.11,2.31)
Friend factor	0.86	0.20	4.23	1.9e-05	(0.28, 0.62)

Figure 5 plots the cost function (Equation 1) averaged over all the workers for the following scenarios: (1) being oblivious to the social effect and considering only the 5 intrinsic features described above, and (2) additionally incorporating the effect of social ties by including the additional *friendship* feature. As we can observe, as λ_r increases, the cost function on the test set reduces and converges. More importantly, the cost function for the 'socially-augmented' model (that includes the *friendship* feature) is lower by $\approx 31\%$ (for our chosen value of $\lambda(r) = 0.1$), compared to the cost function for the "intrinsic-only" model. This result provides evidence that worker preferences on task selection and execution in mobile crowd-sourcing are indeed significantly influenced by (more precisely, *correlated with*) the preferences of other social peers.

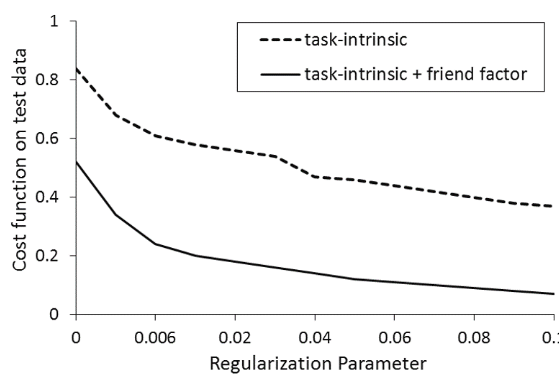


Figure 5. Prediction Cost of Worker Task Preference (with & without social effects)

Key Takeaway: Our investigations show that the user preferences, for different tasks offered on a mobile crowd-sourcing platform, can be modeled quite well as a combination of intrinsic factors (with the popularity of task locations and task rewards being the two most influential factors). However, the preference model is significantly more accurate when one modifies the location-specific preference to take into account the appropriately weighted preferences of peer workers. Similar to observations made on location-based social networks, these results provide further reinforcement that worker choices for tasks in mobile crowd-sourcing platforms exhibit the *homophily* effect.

THE POWER OF COLLABORATIVE REFERRALS

Generally speaking, collaborative work is known to result in superior outcomes (compared to individual efforts) across a variety of domains. The strong social homophily effects, which we have observed in the spatio-temporal patterns of mobile crowd-sourcing, however, call into question the likely success of a collaborative strategy for completing mobile crowd-sourcing tasks. The proof, of course, is in the eating: as shown in Table 1, **referred tasks did achieve a very high completion rate**(with nominees performing 100% of tasks from same-building bundles and over 50% of tasks from multi-building bundles).

We now study this dynamic task referral feature (which was introduced in the *TA$Ker* App over weeks 6-7 of our study) in greater detail. We specifically focus on week 7, as week 6 was a transient period with minimal use of this feature (we used week 6 to advertise and explain this newly introduced feature). Our goal is to try and understand the patterns of tasks offloaded and accepted by the nominator and nominee respectively, and to thereby understand why the referral process works. During the last one week of our study, workers exchanged a total of 85 friend requests. Moreover, 300 tasks (primarily belonging to bundles, but also consisting of individual tasks) were offloaded to such nominees. Each user is allowed to offload (as a nominator) or accept and perform (as a nominee) at most 4 tasks, in a 3-hour time window. The nominee has the flexibility to "reject" a task if he thinks it's not feasible for him to perform it.

Our two key primary findings are:

- *Nominators* always *offload the tasks to their strongest ties.* in particular, for each nominating user, we found that he/she always offloaded tasks to a single nominee–the one who had the strongest tie-strength among the nominator's peers (within the set of *TA$Ker* users). This validates our assumption that collaboration in such mobile crowd-sourcing platforms primarily occur among close friends.

- *Nominators offload tasks primarily because they are too far away.* By comparing the real trajectory of the nominator to the location of the offloaded task, we found that all offloaded tasks would incur a detour of at least 11.75 minutes detour to the nominator (if he/she had actually performed the task). Accordingly, we confirm our initial conjecture that offloading would primarily be used to help complete tasks that were too far from the worker's expected trajectory.

Detour Sensitivity & Task Referrals

To further investigate the phenomenon, we investigate the detours experienced by workers (for performing non-referred vs. referred tasks), and identified two different classes of workers: (1) *Detour Sensitive*: these are workers who always accept and perform tasks that are close by (i.e., are either on the same floor or in the same building and thus result in very low detour), even if the task reward is relatively lower; and (2) *Detour Insensitive*: these are workers who are willing to take longer detours in order to earn more (they can be viewed as similar to super-agents). It is worth stating that these two classes had approximately the same number of "push" and "pull" users–i.e., detour sensitivity was not a function of the recommendations made by the *TA$Ker* platform.

Nominee Perspective: Figure 6 plots the average detour incurred by various nominees across their completed tasks: the x-axis plots the average detour for non-referred tasks during weeks 1-5, while the y-axis plots the average detour for all referred (offloaded) tasks during week 7. We can easily separate the cluster of detour-sensitive nominees (those who previously performed only tasks with average detour ≤ 5 mins) from the detour-insensitive ones (those whose average detour was higher than 10 mins–i.e., spanned multiple buildings). We see that detour-insensitive workers experience no difference between non-referred vs. referred tasks–they continue to experience detours of >10 mins (equivalent to visiting a building that is two blocks away).

In contrast, detour-sensitive nominees showed a dramatic change in their pattern: for refereed tasks that they completed, they experienced detours that are more than twice what they normally tolerate. This discrepancy explains why task offloading is successful: even though the nominee is likely to be far away from the offloaded task, the nominee ends up completing the task (irrespective of her detour sensitivity). To mathematically verify this change, we conducted a $t-$test on the average detour (for such detour-sensitive users) between referred vs. non-referred tasks, and confirmed that the differences were statistically significant (confirmed by a t-test with p-values < 0.0001).

Nominator Perspective: Figure 7 plots the similar detour values for various nominators: the x-axis plotting the average detour for non-referred tasks during weeks 1-5, with the y-axis plotting the average detour for all referred (offloaded) tasks during week 7. We see the two categories of users: detour-sensitive users have x-axis values less than 5 mins, while detour insensitive users have x-axis values of > 10 mins. The interesting observation is that the detours associated with the referred tasks is usually high (> 10 mins) for either category of workers.

For detour-sensitive workers, this implies an inherent willingness to initially accept tasks with longer detour (the nominator must first accept the task before she can offload it). Apparently, the availability of the task referral feature enables such workers to accept more out-of-the-way tasks, based on the expectation that these tasks can then be offloaded (if needed) to their friends. To mathematically verify this change, we conducted a $t-$test on the average detour (for such detour-sensitive nominators) between referred vs. non-referred tasks, and confirmed that the differences were statistically significant (p-values < 0.0001).

Impact of Tie-Strength on Detour

Our investigations suggest that the success of dynamic collaboration (i.e., peer referrals) is primarily driven by the tie-strength between the nominator and nominee, and not due

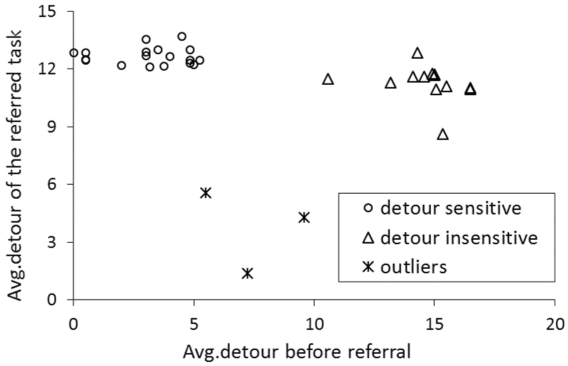


Figure 6. Average detour (Nominee) before and after task offloading feature is introduced

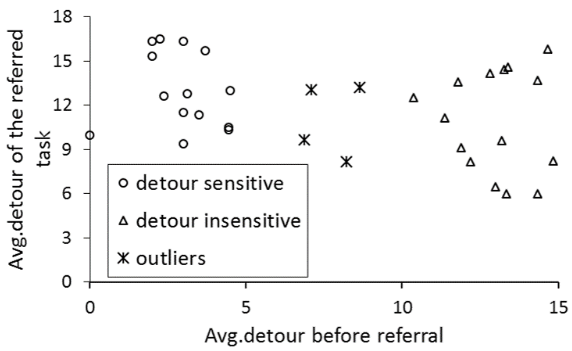


Figure 7. Average detour (Nominator) before and after task offloading feature is introduced

to any possible reductions in detour caused by the referral. To further quantify this effect, we looked at both detour-sensitive and detour-insensitive *nominees* and computed (just for week 7), the average detours that they experienced (over all completed tasks) for non-referred and referred tasks. Figure 8(a) plots these average detour for detour-sensitive nominees, while Figure 8(b) plots these average detours for detour-insensitive nominees. The figures further reinforce our central observation: while detour-insensitive users accept referred tasks without any noticeable change in their detour overheads, detour-sensitive users accept tasks even though the added overhead is noticeably larger.

For any particular user i, let d_i denote the difference between these two average detour values. We finally investigate if this value of d_i is itself influenced by the degree of the friendship between the nominator and the nominee. In particular, we investigate the correlation between d_i (the *additional* detour overhead) and the tie-strength between the nominator k and nominee i (denoted by t_{ki}). The resulting Spearman correlation coefficient is 0.64, indicating that this *willingness to take on additional detour is strongly influenced by the strength of the friendship between the (nominator, nominee) pair.*

Key Takeaway: Our careful analysis of the differences in detours between non-referred and refereed tasks shows that the peer referral process works even though the task-related detour experienced by the nominee is appreciably high (greater than 10 mins, on average). While ≈ 50% of workers are just intrinsically insensitive to such high detours, the other 50% are quite sensitive. However, they also accept and complete such referred tasks, principally because the referral request comes from their close friends. Moreover, we show that: higher the tie-strength between the (nominator, nominee) pair, the greater the willingness of the nominee to incur longer detours for the referred tasks. *Clearly, future mobile crowd-sourcing platforms should enable such selective task offloading to close friends, to increase both the acceptance and completion rate of tasks.*

DISCUSSION

Many crowd-sourcing efforts presume humans as independent sources of information. Many routine uses of crowd-sourcing, such as surveys and behavioral experiments depend on one crowd-worker's response, assuming that he is not colluding with peers. Also, the platforms often bias the output obtained from a crowd-worker by considering the majority votes [28], but if all the crowd-workers' responses are correlated, then the results can be skewed [20]. However, our study empirically shows that workers themselves are putting collaboration back into such practices. They in fact collaborate with their peers dynamically and increased the task completion rate.

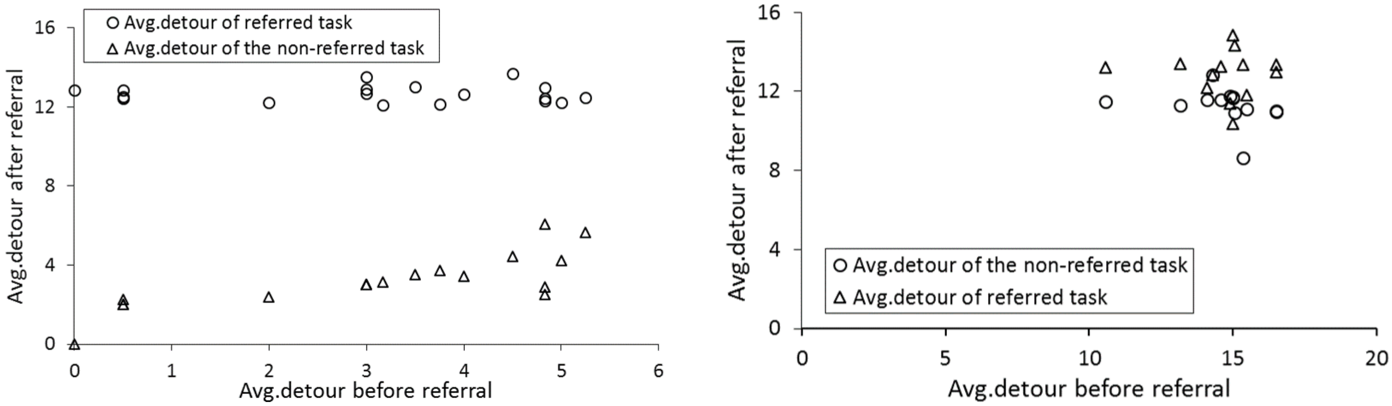
In this paper we presented a phenomenon called *task completion homophily*: workers with strong strength-of-ties tend to have high similarity in their spatio-temporal patterns of task execution. We show that there is a strong correlation (Pearson correlation value of 0.74) between the tie-strength of a worker pair and their spatio-temporal task completion patterns. This corroborates with the findings reported in [7], where they use location-based social network data and cellular location records to demonstrate social relationship, in particular the visit pattern of friends. Furthermore, as reported in [9], a user's movement behavior could be more accurately predicted by factoring in the movements of her friends.

Recently, online crowd-sourcing research community focused on the value of coordinating human interaction at scale. The notion of "social facilitation" [8, 12, 13] is well studied in the online community. Such studies offer a valuable case study for incorporating one's social network into crowd-sourcing processes [2]. In contrast to the research that primarily looks at online crowd-sourcing scenarios, in this paper, we show that in mobile crowd-sourcing, incorporating social factors (specifically, the strength-of-tie weighted location preference of all other peer workers) in a per-user task preference model results in a significant improvement, increasing the prediction accuracy by 31%.

Limitations and Challenges

Our work and results provide compelling evidence that the incorporation of a "dynamic peer-to-peer offloading" capability can significantly improve the efficacy of urban crowd-sourcing. There are, however, several open issues and challenges that need further investigation:

- *Applicability to City-Scale Crowd-sourcing: TA$Ker* is a campus-scale crowd-sourcing platform. Consequently, all



(a) Detour Sensitive Nominee (b) Detour Insensitive Nominee

Figure 8. Average detour of the referred and non-referred tasks of (a) detour-sensitive, and (b) detour in-sensitive users

the workers are long-term residents of the campus and form long-term social linkages; moreover, their on-campus movement patterns will naturally have high overlap. Such social ties or movement overlap may not be as pronounced in a city-wide deployment of crowd-sourcing; close friends, for example, may live in entirely different neighborhoods and have no overlap in commuting paths. The efficacy of such task offloading in such scenarios is not immediately obvious. However, it is possible that task execution may indeed have *locality-of-reference*–e.g., such offloading may still be useful within neighborhoods (or housing estates), where neighbors have varying degrees of social ties and are likely to have appreciable spatial overlap. Our current belief is that such dynamic offloading may indeed prove effective for *neighborhood-scale* crowd-sourcing (e.g., when used to solicit reports on municipal resources such as garbage bins and street lights), as the participants in such a platform would have high spatiotemporal overlap as well.

- *Task Corroboration and User Reputation:* Truth discovery & corroboration of results is a key requirement for reliable urban crowd-sourcing. Present mechanisms for task corroboration often rely on individual-specific reputation measures, which are appropriate for scenarios where workers perform tasks independently (interacting only with the crowd-sourcing platform). If task offloading and shared execution becomes the norm, such corroboration measures may need to be redesigned to better take into account the social ties and interactions among task nominators and nominees (are workers less or more likely to generate spurious reports when they're asked to do so by their friends?).
- *Design of Incentive Sharing:* Our work in this paper has adopted a natural, but simplistic, incentive sharing scheme, where the task reward is equally shared among the nominator and nominee. It is entirely possible that the incentive sharing schemes may need to be redesigned to better harmonize with the different price sensitivities of different workers. For example, would a nominator be willing to refer the task to a friend whose profile indicates that she demands a higher fraction of the reward, due to possibly facing a higher detour overhead? Moreover, in our current studies, the task execution interval (T_s, T_e) is specified independently of the task's reward and whether it is performed by the nominator or a nominee. In future, it is possible that tasks may have a more flexible rewards vs. execution interval curve, which may provide workers with more interesting choices between performing the task in a more-delayed fashion (thus earning less but receiving the entire reward) or offloading it for more immediate execution (earning a higher but shared reward).
- *Lack of User Feedback Data:* Unfortunately, during our study period, we did not explicitly survey the participants to find their opinions about the usefulness of this referral feature. We plan to conduct such explicit surveys during ongoing/future studies, to help us better understand additional collaborative capabilities that they may desire.

CONCLUSIONS AND FUTURE WORK

To provide better understanding of dynamic collaboration among peer crowd workers in mobile crowd-sourcing, we have conducted extensive real-world (over 7 weeks and with 1300 workers) using *TA$Ker* – a campus based mobile crowd-sourcing platform. In this study, collaboration is characterized by the act of peer referrals, where a worker who initially committed to perform a task, but is unlikely to reach the task location, can offload the task to his friend. We first show the existence of a significant "homophily" effect in the execution of mobile crowd-sourcing tasks in campus. Though, at first glance, this "homophily" feature implies that the explicit collaboration among "friend " crowd workers may not be useful, we show later that peer assistance outweighs the effect of "homophily" in mobile crowd-sourcing platforms.

Our empirical analysis reveals the following:

- The physical tie-strength of a crowd worker pair and their spatio-temporal task completion pattern are strongly correlated (with Pearson correlation of 0.74), exhibiting "task completion homophily" among *TA$Ker* workers.
- Incorporating social factors in a per-user preference model yields more accuracy in prediction compared to the one only with task-intrinsic properties. In particular, we show

that incorporating a social attribute called the "friendship score" (i.e., the strength of tie weighted location preference of all other peer workers) improved the per-user task preference accuracy by 31%, compared to a model that used only task-intrinsic properties such as incentive, popularity of the task location and task complexity.

- Dynamic peer referrals is very successful: its introduction increased the overall task completion rate by 14%. Moreover, nominees exhibit extremely high compliance to such offloading requests–performing 100% of requests for tasks in spatially-contained bundles and over 50% of requests for geographically-dispersed bundles. Over a 1 week trial period (happen to be the university's exam week), the task referral feature is used by at least 10% of the active workers of *TA$Ker* to offload around 300 tasks.
- A nominee's willingness to tolerate a higher detour offloaded task is strongly correlated (Pearson correlation 0.64) to the physical tie strength between the nominee and nominator.

In ongoing and future work, we shall systematically use *TA$Ker* to study the interplay between such dynamic task offloading and other crowd-sourcing parameters (such as incentive sharing strategies).

Acknowledgement

This material is supported partially by the Singapore National Research Foundation, under its International Research Centre Singapore Funding Initiative and administered by the IDM Programme Office, and partially by the Air Force Research Laboratory, under agreement number FA2386-14-1-002. The view and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the US Government.

The authors would like to humbly thank Nikita Jaiman and Ming Hui Desmond Koh for developing the *TA$Ker* system and providing valuable support regarding this work.

REFERENCES

1. Florian Alt, Alireza Sahami Shirazi, Albrecht Schmidt, Urs Kramer, and Zahid Nawaz. 2010. Location-based Crowdsourcing: Extending Crowdsourcing to the Real World. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. 13–22.
2. Michael S. Bernstein, Densey Tan, Greg Smith, Mary Czerwinski, and Eric Horvitz. 2010. Personalisation via Friendsourcing. *ACM Transactions on Computer Human Interaction* 2, 17 (2010), 1–28.
3. S. Boyd and L. Vandenberghe. 2004. Convex Opimization. *Cambridge Universit Press* (2004).
4. Jon Chamberlain. 2014. Harnessing the intelligence of the crowd for problem solving and knowledge discovery. In *AAAI conference on Human Computation and Crowdsourcing*.
5. Cen Chen, Shih-Fen Cheng, Aldy Gunawan, Archan Misra, Koustuv Dasgupta, and Deepthi Chander. 2014. TRACCS: Trajectory-Aware Coordinated Urban Crowd-Sourcing. In *2nd AAAI Conference on Human Computation and Crowdsourcing*. 30–40.
6. Cen Chen, Shih-Fen Cheng, Hoong Chuin Lau, and Archan Misra. 2015. Towards city-scale mobile crowdsourcing: Task recommendations under trajectory uncertainties. In *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-15)*. 1113–1119.
7. Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. 1082–1090.
8. Franco Curmi, Maria Angela Ferrario, John Whittle, and Floyd Mueller. 2016. Crowdsourcing Synchronous Spectator Support (Go on, Go on, you're the best). In *SIGCHI Conference on Human Factors in Computing Systems*. 757–766.
9. Manlio De Domenico, Antonio Lima, and Mirco Musolesi. 2013. Interdependence and Predictability of Human Mobility and Social Interactions. *Pervasive Mob. Comput.* 9, 6 (Dec. 2013), 798–807.
10. M. D. Gordon. 1980. A critical reassessment of inferred relations between multiple authorship, scientific collaboration, the production of papers and their acceptance for publication. *Scientometrics* 2 (1980), 193–201.
11. Mary L. Gray, Siddharth Suri, Syed Shoaib Ali, and Deepti Kulkarni. 2016. The Crowd is a Collaborative Network. In *The 19th ACM conference on Computer-Supported Cooperative Work and Social Computing*. 134–147.
12. Shih Wen Huang and Wan Tat Fu. 2013a. Don't hide in the crowd! Increasing social transparency between peer workers improves crowdsourcing outcomes. In *SIGCHI Conference on Human Factors in Computing Systems*. 621–630.
13. Shih Wen Huang and Wan Tat Fu. 2013b. Motivating crowds using social facilitation and social transparency. In *The 16th ACM conference on Computer-Supported Cooperative Work and Social Computing*. 149–152.
14. Brent Hecht Jacob Thebault-Spieker, Loren Terveen. 2015. Avoiding the South Side and the Suburbs: The Geography of Mobile Crowdsourcing Markets *(CSCW '15)*. 265–275.
15. Kasthuri Jayarajah, Archan Misra, Xiao-Wen Ruan, and Ee-Peng Lim. 2015. Event Detection: Exploiting Socio-Physical Interactions in Physical Spaces. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 508–513.

16. Azeem J. Khan, Vikash Ranjan, Trung-Tuan Luong, Rajesh Krishna Balan, and Archan Misra. 2013. Experiences with performance tradeoffs in practical, continuous indoor localization.. In *IEEE WOWMOM*. 1–9.

17. Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The Future of Crowd Work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. 1301–1318.

18. Munir Mandviwalla and Lorne Olfman. 1994. What do groups need? A proposed set of generic groupware requirements. *ACM Transactions on Computer Human Interaction* 1, 3 (1994), 245–268.

19. Archan Misra and Rajesh Krishna Balan. 2013. LiveLabs: Initial Reflections on Building a Large-scale Mobile Behavioral Experimentation Testbed. *SIGMOBILE Mobile Computing and Communications Review* 17, 4 (2013), 47–59.

20. Lev Muchnik, Sinan Aral, and Sean J. Taylor. 2013. Social Influence Bias: A Randomized Experiment. *Science* 341, 6146 (2013), 647–651.

21. M. Mushthag and D. Ganesan. 2012. The role of superagents in mobile crowdsourcing. *Workshops at AAAI conference on Human Computation and Crowdsourcing* (2012).

22. Mohamed Musthag and Deepak Ganesan. 2013. Labor dynamics in a mobile micro-task market. In *SIGCHI Conference on Human Factors in Computing Systems*. 641–650.

23. T. Okada and H. A. Simon. 1997. Collaborative discovery in a scientific domain. *Cognitive Science* 21 (1997), 109–146.

24. O. Persson, W. Ganzel, and R. Danell. 2004. Inflationary bibliometric values: The role of scientific collaboration and the need for relative indicators in evaluation studies. *Scientometrics* 60 (2004), 421–432.

25. Rijurekha Sen, Youngki Lee, Kasthuri Jayarajah, Archan Misra, and Rajesh Krishna Balan. 2014. GruMon: fast and accurate group monitoring for heterogeneous urban spaces. In *12th ACM Conference on Embedded Network Sensor Systems*. 46–60.

26. Asron Shaw and Benjamin Mako Hill. 2014. Laboratories of Oligarchy? How the iron law extends to peer production. *Communications* 64 (2014), 214–238.

27. J. Singh and L. Fleming. 2009. Lone inventors as sources of breakthroughs: Myth or reality? *Management Science* 56 (2009), 41–56.

28. Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and Fast – But Is It Good? Evaluating Non-expert Annotations for Natural Language Tasks. In *The ACL conference on Empirical Methods in Natural Language Processing (EMNLP)*. 254–263.

29. P. Thagard. 1997. Collaborative knowledge. *Nous* 31 (1997), 242–261.

30. Jing Wang, Siamak Faridani, and Panagiotis Ipeirotis. 2011. Estimating the Completion Time of Crowdsourced Tasks Using Survival Analysis Models. In *Workshop on Crowdsourcing for Search and Data Mining (CSDM)*. 31–34.

31. Haiyi Zhu, Robert Kraut, and Aniket Kittur. 2012. Organizing without formal organization: Group Identification, goal setting and social modeling in directing online production. In *The 15th ACM conference on Computer-Supported Cooperative Work and Social Computing*. 935–944.